IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Matthew Poppa
May 25th 2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data Collection using API
  - Data Collection by Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Map created with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive Analysis with Folium and Plotly Dash
  - Predictive results

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. The goal of this project is to obtain data about Falcon 9 rocket launches and use machine learning to predict if the first stage will land safely to be reused.
- Problems we want to find answers to
  - Which variables have an affect on the success rate of the first stage landing safely?
  - How do each of these variables change the success rate?
  - How do we set these variables so that there is the highest possible safe landing success rate?

Section 1

Methodology

Methodology

Executive Summary

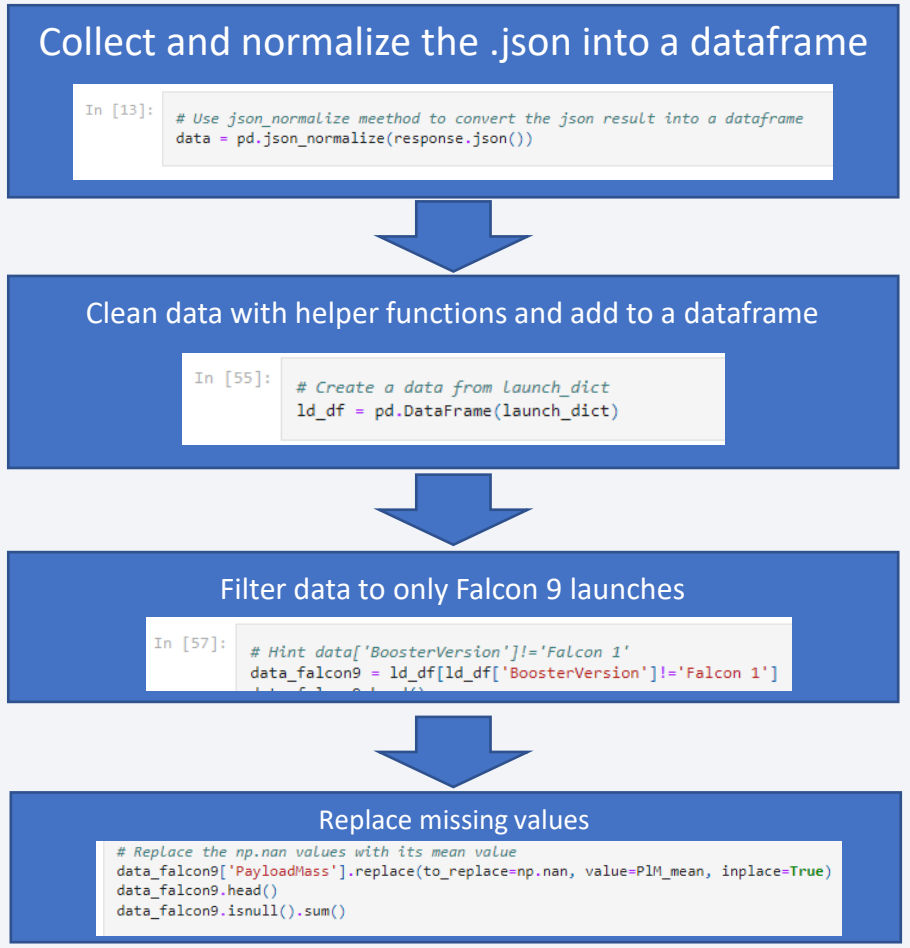
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropped unnecessary columns and applied one-hot-encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data collection was accomplished through the SpaceX REST API and Web Scraping data from the tables of SpaceX's Wikipedia page.
- Data from the SpaceX REST API was obtained as a .json file and then converted into a pandas dataframe using .json_normalize().
- Then the data was stripped down to only the necessary columns, cleaned up using provided helper functions, saved into a new pandas dataframe, filtered down to only Falcon 9 records, and then replacing null or missing values.
- Data from the Wikipedia table was scraped into a BeautifulSoup object.
- The column names were extracted and then the HTML table was parsed to a pandas dataframe for future analysis.

Data Collection – SpaceX API

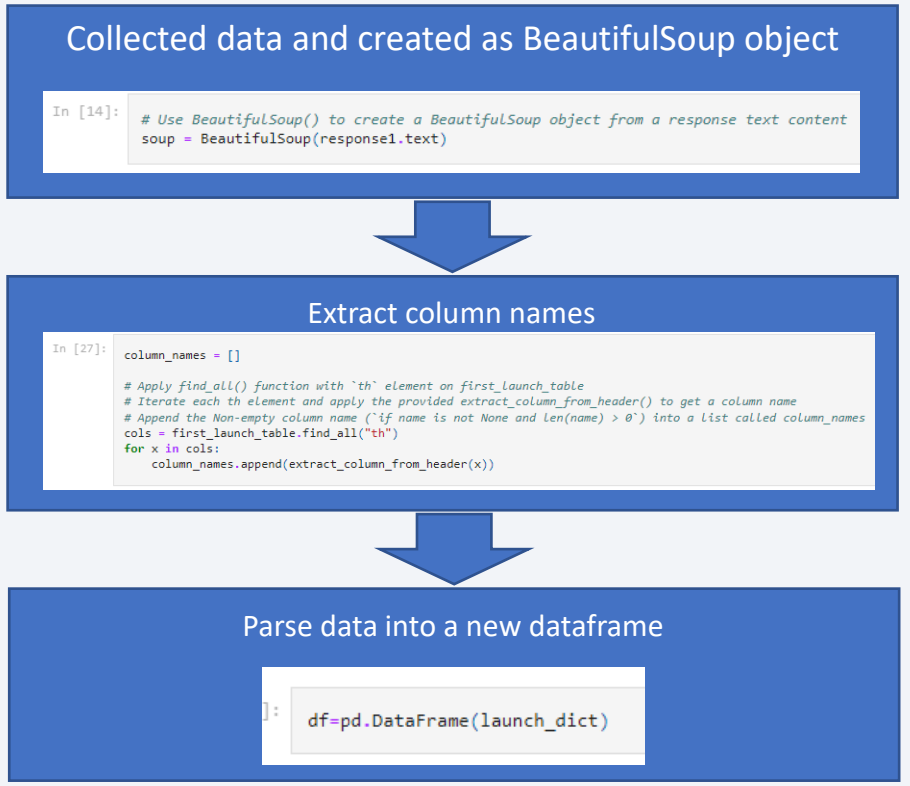
- Data was collected using the public SpaceX REST API. Then processed according to the following flow chart:



Link to Source Notebook on GitHub

Data Collection - Scraping

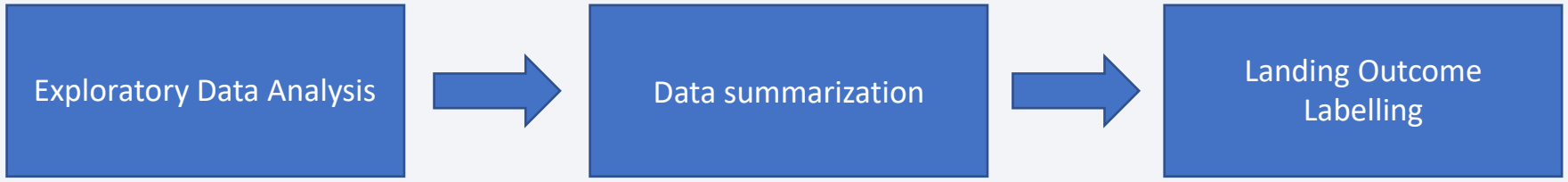
- Data was scraped from the SpaceX Wikipedia page and then processed according to the following flow chart:



[Link to Source Notebook on GitHub]

Data Wrangling

- Initial Exploratory Data Analysis was performed to better understand the data.
  - Summaries:
    - Number of launches for each launch site
    - Occurrences of each orbit
    - Number and occurrence of each mission outcome per orbit type
  - Created label for both possible landing outcomes:
    - Success (1)
    - Failure (0)



Link to Source Notebook on GitHub

EDA with Data Visualization

- To explore the data a number of plots were created:
  - Scatter plots used to visualize the relationship between different variables:
    - Flight Number visualized with Payload Mass
    - Flight Number visualized with Launch Site
    - Payload Mass visualized with Launch Site
    - Flight Number visualized with Orbit type
    - Payload Mass visualized with Orbit type
  - Bar chart to visualize the relationship of success rate with each orbit type.
  - Line plot to visualize the affect of time (by year) on the success rate.

Link to Source Notebook on GitHub

EDA with SQL

- **SQL queries run to explore the data for the following:**
  - Unique launch site names.
  - 5 records with launch sites beginning with “CCA”.
  - Total payload mass carried by boosters launched by NASA (CRS).
  - Average payload mass carried by booster version F9 v1.1.
  - Date of the first successful landing outcome in ground pad was achieved.
  - Names of the boosters which have success in drone ship and have payload mass greater than 4,000 but less than 6,000.
  - Total number of successful missions and failure missions.
  - Names of the booster versions that have carried the maximum payload mass.
  - List of records for the year 2015 with month names, failed landing outcomes for drone ship, booster version, and launch site.
  - Ranked count (in descending order) of successful landing outcomes between June 4th 2010 and March 20th 2017.

Link to Source Notebook on GitHub

Build an Interactive Map with Folium

- Interactive map created with Folium to display the following:
  - Markers for each launch site.
  - Circles highlighting the area around those sites.
  - Marker clusters to display successes and failed launches for each site.
  - Lines to display the distance from a launch site to a coast line and other landmark location.

Link to Source Notebook on GitHub

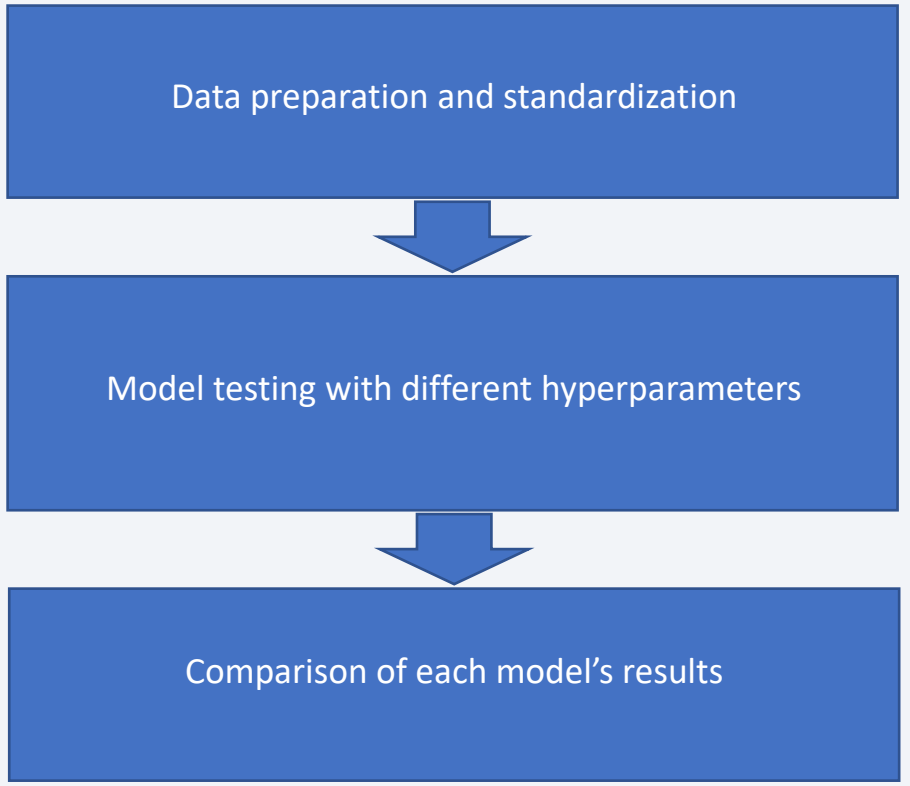
Build a Dashboard with Plotly Dash

- A dashboard created with Plotly Dash to visualize the following data interactively:
  - Pie chart; showing the percentage of successful launches for each site.
  - Scatter plot; showing payload ranges and corresponding success rates.
- Both of these allow the user to quickly and easily select the data they are most interested in. This can be used to find the launch site and payload range with the highest success rate.

Link to Source Notebook on GitHub

Predictive Analysis (Classification)

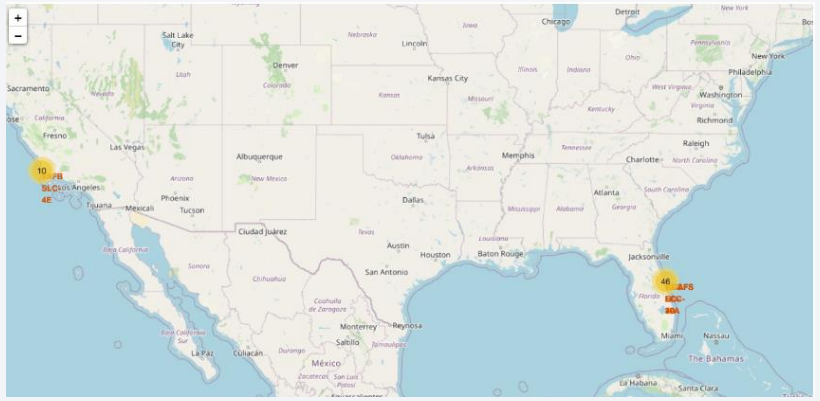
- Prediction through 4 different classification models:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K-nearest Neighbors
- Each was processed according to the flow chart to the right:

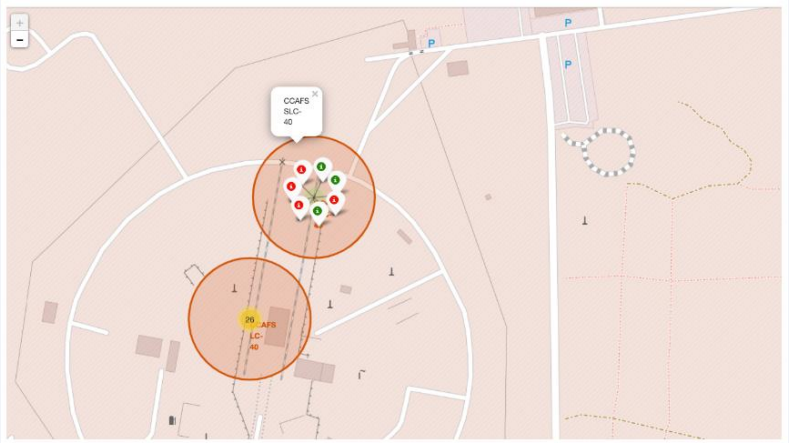


Link to Source Notebook on GitHub

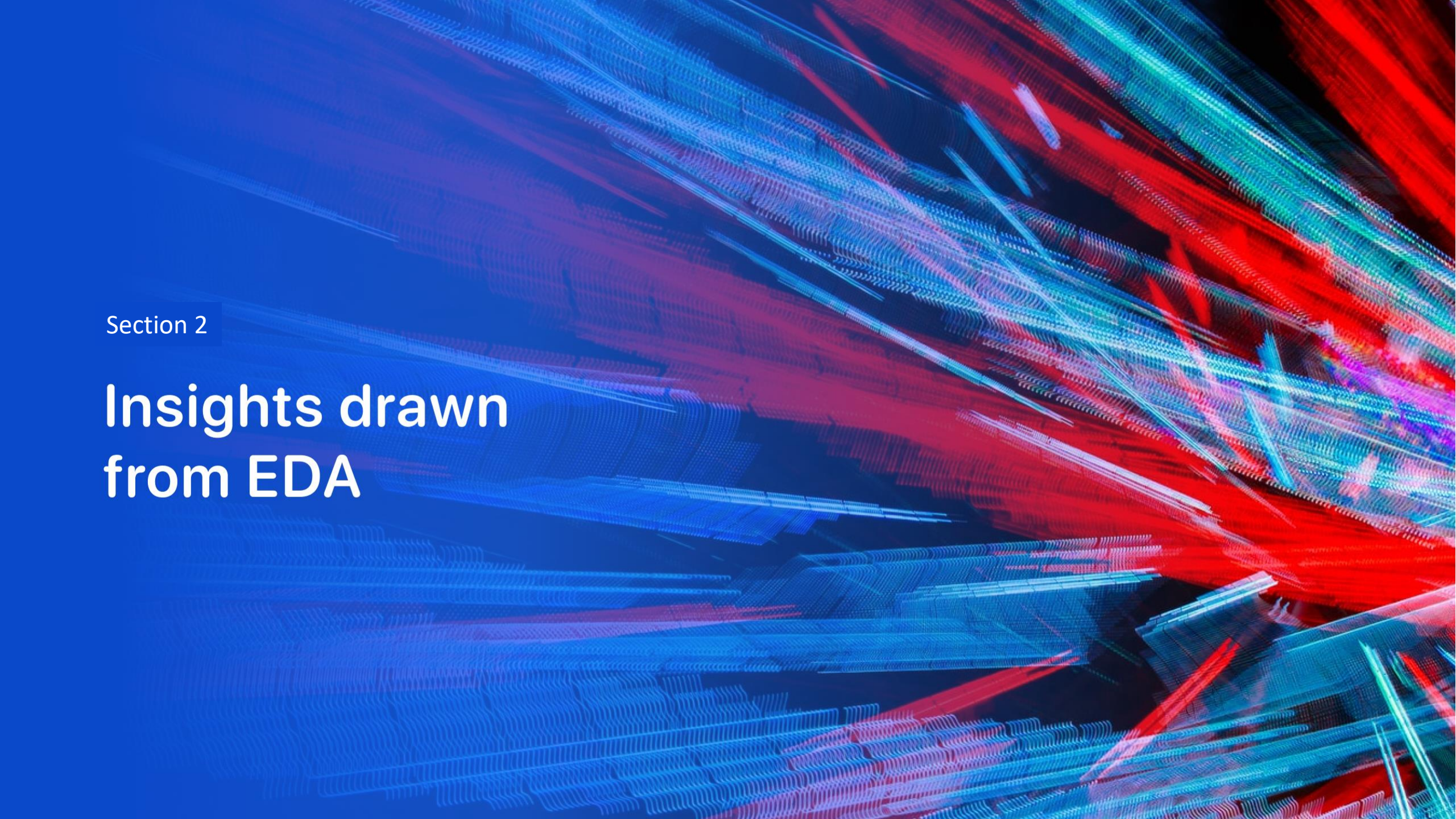
Results

- Exploratory data analysis results:
  - The success rate has increased over the years.
  - The launch site with the highest success rate is KSC LC-39A.
  - The best recovery method is the drone ship.



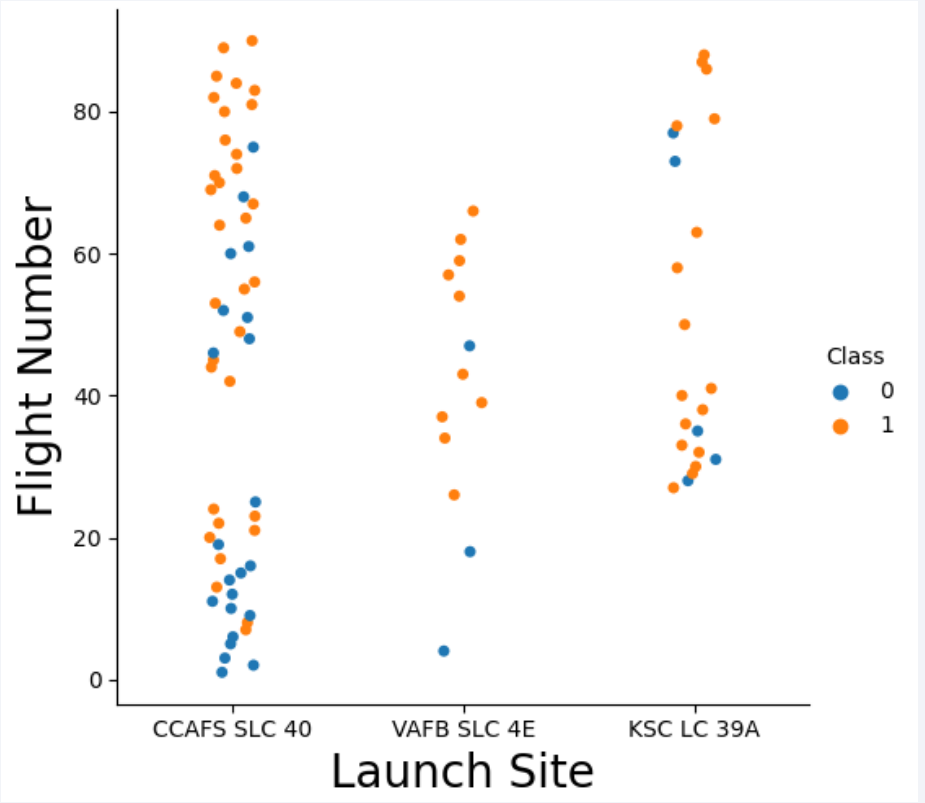


- Predictive analysis results:
  - Found the best hyperparameter values for each model.
  - Determined that at those hyperparameters each model performed relatively the same on the test data.


Section 2
Insights drawn from EDA

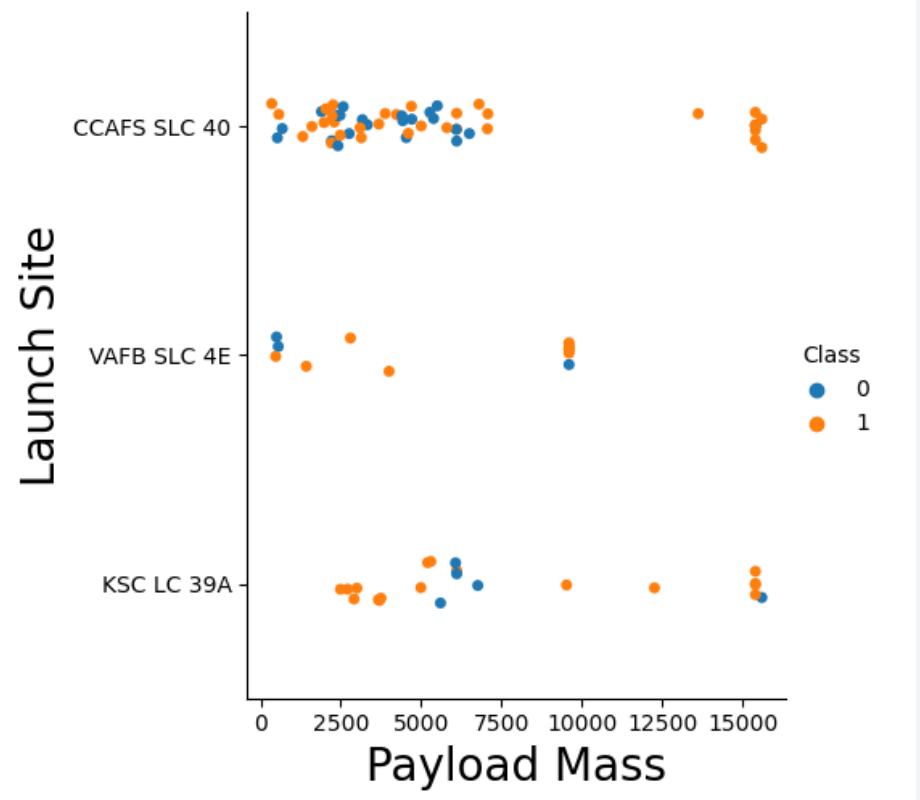
Flight Number vs. Launch Site

- The success rate has increased over time.
- Launch site CCAFS SLC 40 has most of the recent successful landings.



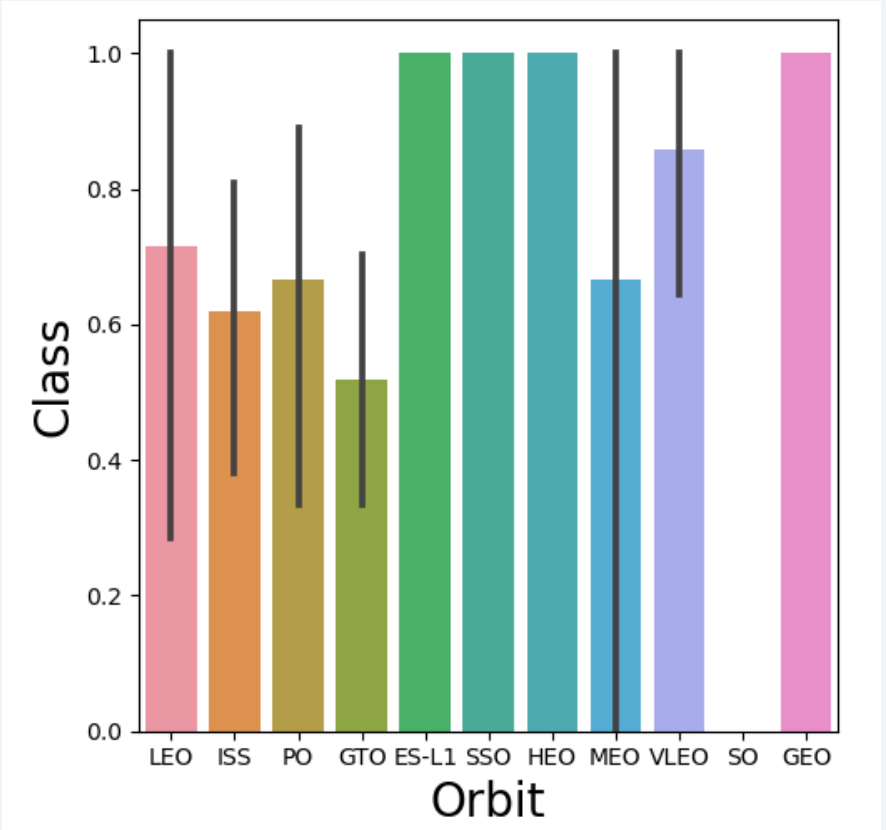
Payload vs. Launch Site

- Payloads of 9,000kg or more have higher success rates.
- Payloads above 10,000kg have not been tested at VAFB SLC 4E.



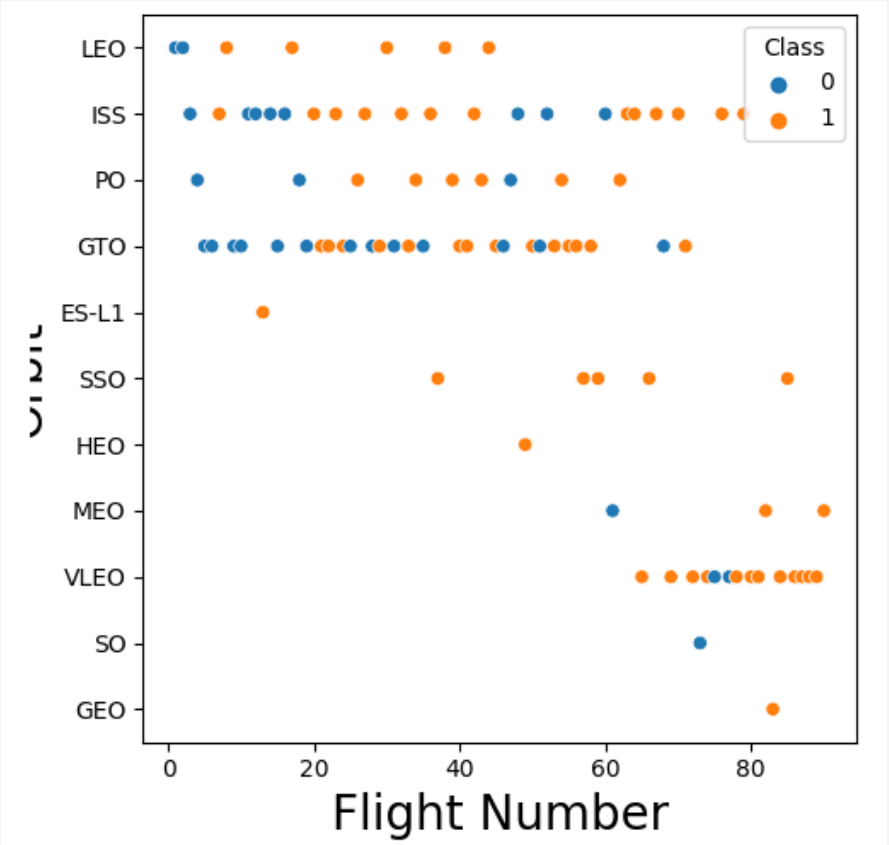
Success Rate vs. Orbit Type

- Orbit types ES-L1, SSO, HEO, and GEO have the highest success rates.
- Orbit type SO has no successes.



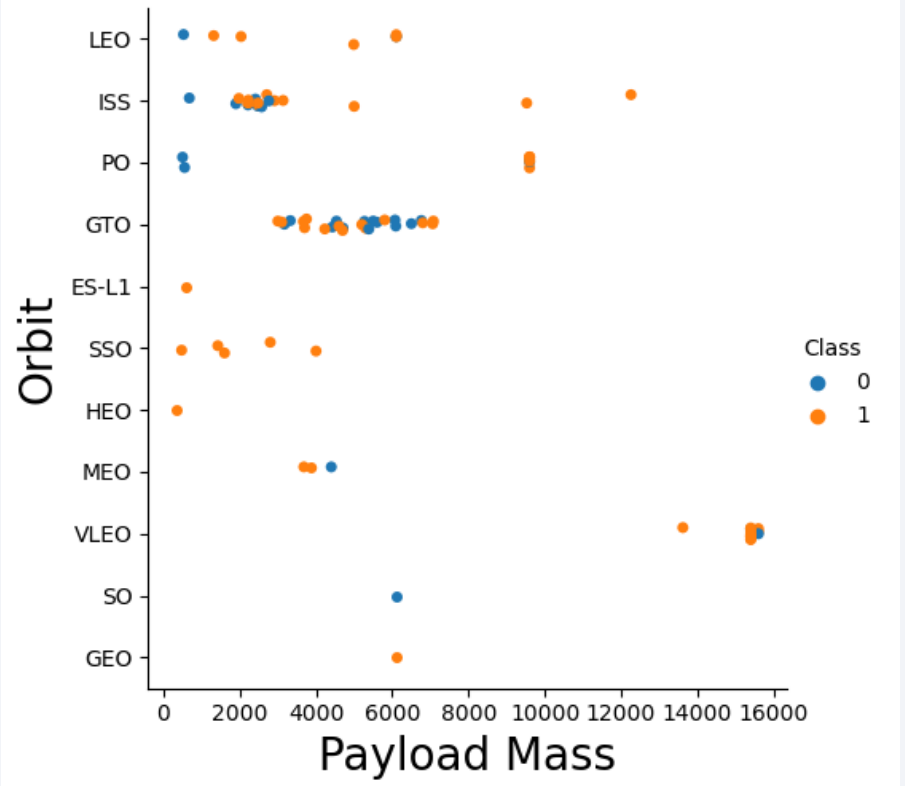
Flight Number vs. Orbit Type

- Success rates for all Orbit types increase over time.
- The success rate for GEO, SO, HEO, and ES-L1 are skewed due to each only having one flight.



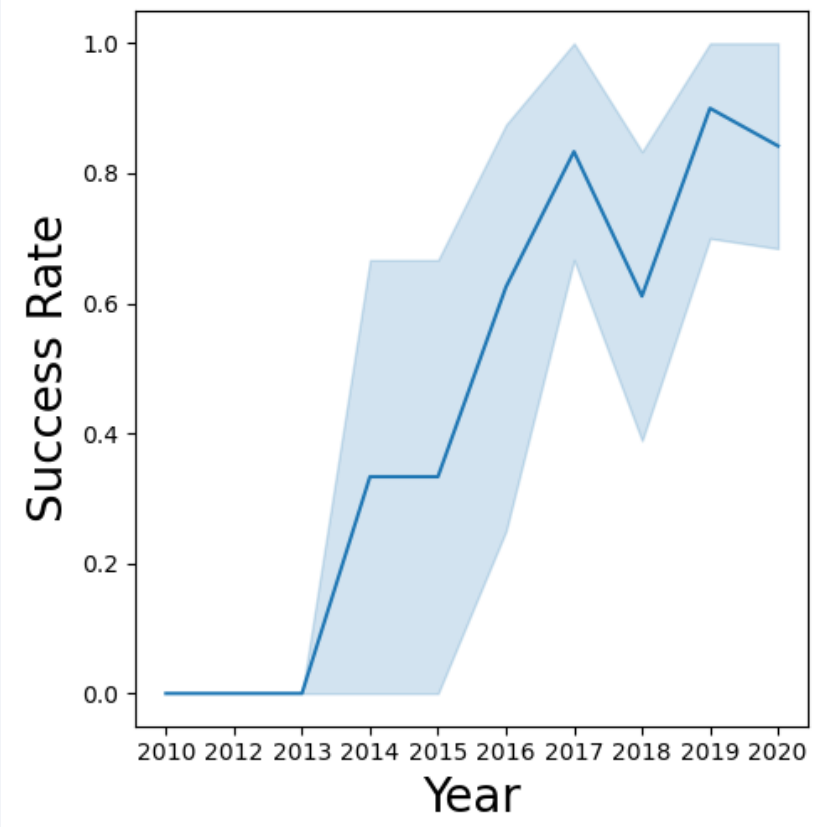
Payload vs. Orbit Type

- Flights with payload mass of over 9,000kg only occurred with orbit type PO, ISS, and VLEO.



Launch Success Yearly Trend

- The success rate has significantly improved over the years with a small set back in 2018 before returning to new highs.



All Launch Site Names

- Using the DISTINCT command on Launch_Site only unique launch site names are displayed.

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- Using the LIKE and LIMIT 5 commands within the WHERE clause on Launch_Site the first 5 records are shown that have a launch site beginning with “CCA”
- The % symbol within ‘CCA%’ tells the query to search for launch sites beginning with CCA and not just ones that are exactly CCA.

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Using the SUM() command on PAYLOAD_MASS__KG_ and a WHERE clause to use only the customers named ‘NASA (CRS)’ the query will display the total payload mass for all launches for NASA (CRS).

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Using the AVG() command on PAYLOAD_MASS__KG_ and with a WHERE clause to use only records with a booster version of ‘F9 v1.1’ the query returns the average payload mass for booster version F9 v1.1.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Using the MIN() command on Date and a WHERE clause to use only mission outcomes that were successful the query returns the earliest date of a successful mission. In other words, the very first successful landing.

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Mission_Outcome = 'Success'
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
01-03-2013

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using the command DISTINCT on Booster_Version and a WHERE clause to use landing outcomes with string of 'Success (drone ship)' plus pay load mass above 4000 and below 6000 the query returns a list of the boosters that have had successful landings with drone ships and have a payload between 4000 and 6000.

```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Using the COUNT() command on Mission_Outcome and a GROUP BY clause to group by the mission outcome types the query returns a list with the count of each mission outcome type.

```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTBL GROUP BY Mission_Outcome
```

```
* sqlite:///my_data1.db
Done.
```

Mission_Outcome	COUNT(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Using Booster_Version in the SELECT clause and a WHERE clause to use the maximum payload size only, determined by the MAX() command on PAYLOAD_MASS__KG_, the query returns all booster versions that have carried that largest payload size.

```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Using substr(Date, 4, 2) in the SELECT clause and a WHERE clause to use only landing outcomes of ‘Failure (drone ship)’ plus substr(Date, 7, 4) = ‘2015’ the query returns the months for each failed drone ship landing in the year 2015.

```
%sql SELECT substr(Date, 4, 2), "Landing _Outcome", Booster_Version, Launch_Site FROM SPACEXTBL WHERE "Landing _Outcome" = '
```

```
 * sqlite:///my_data1.db
Done.
```

substr(Date, 4, 2)	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Using a WHERE clause for dates between 2010-06-04 and 2017-03-20 and landing outcomes with success in the name (through the LIKE command). Then using an GROUP BY clause to group by the date. Finally an ORDER BY clause to have the dates in descending order. The query then returns the list of dates with successful landing outcomes in order from oldest to newest as seen here.

```
%sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXTBL \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

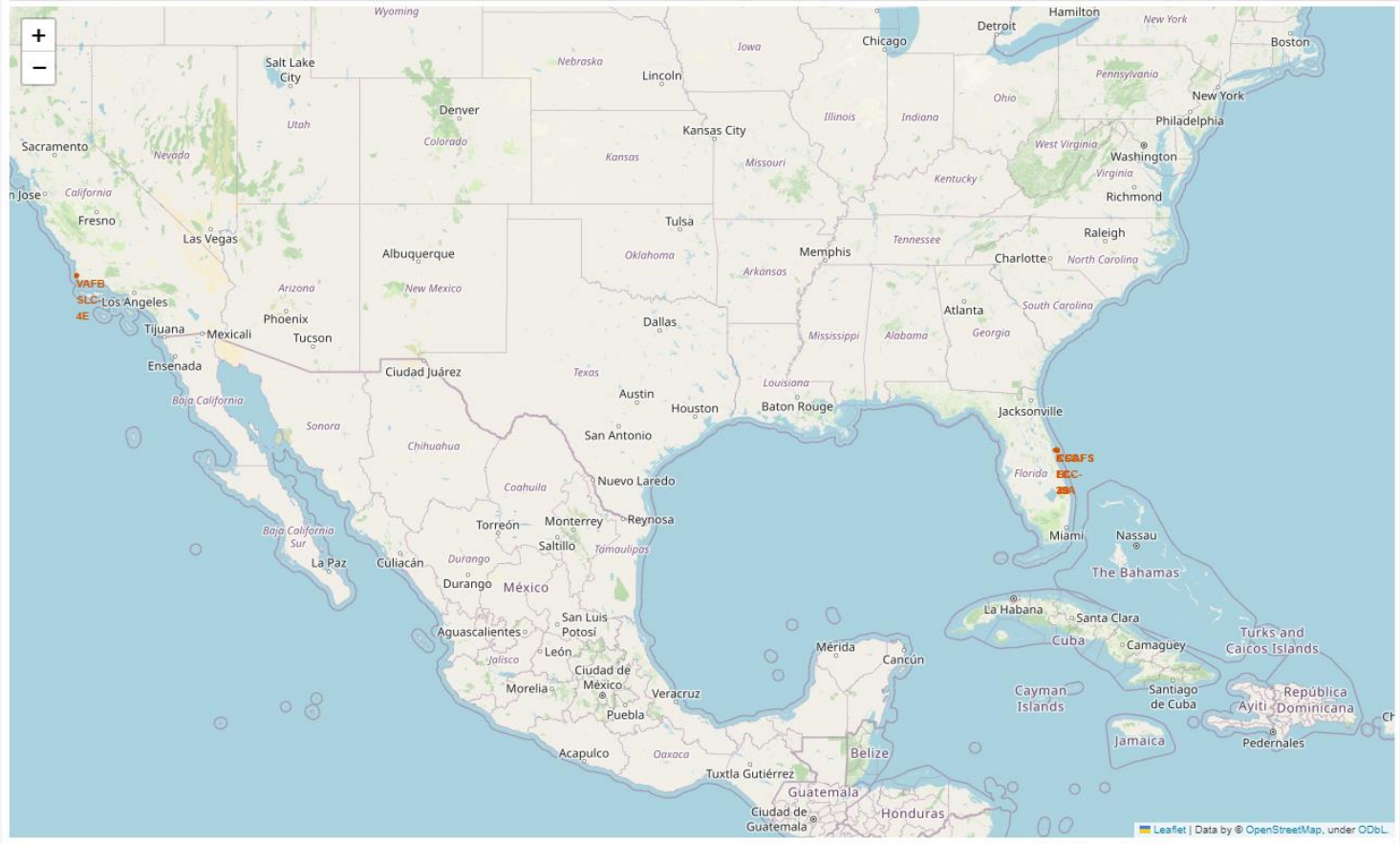
DATE	COUNT
2015-12-22	1
2016-04-08	1
2016-05-06	1
2016-05-27	1
2016-07-18	1
2016-08-14	1
2017-01-14	1
2017-02-19	1

Section 3

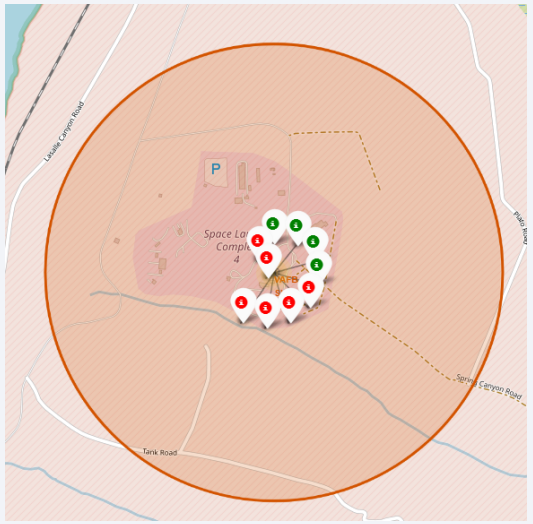
Launch Sites Proximities Analysis

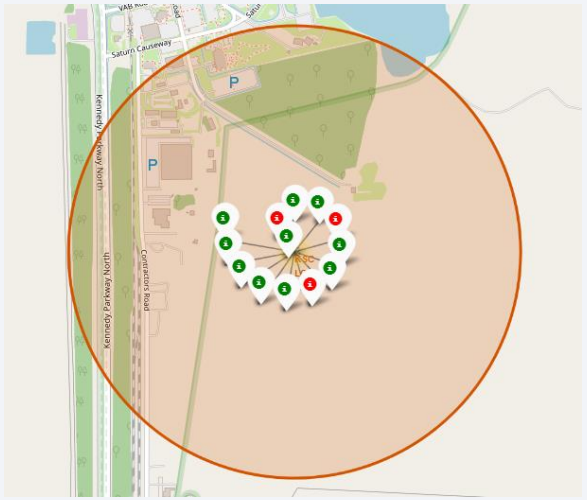
SpaceX Launch Site Locations Visualized

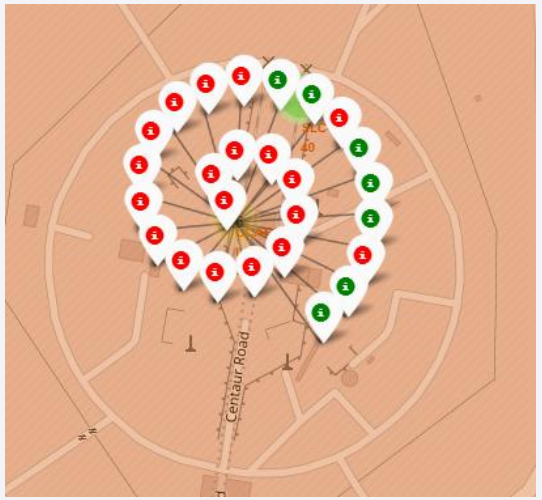
- Each launch site is located on the coast within the United States of America.



Launch Results Visualized by Color



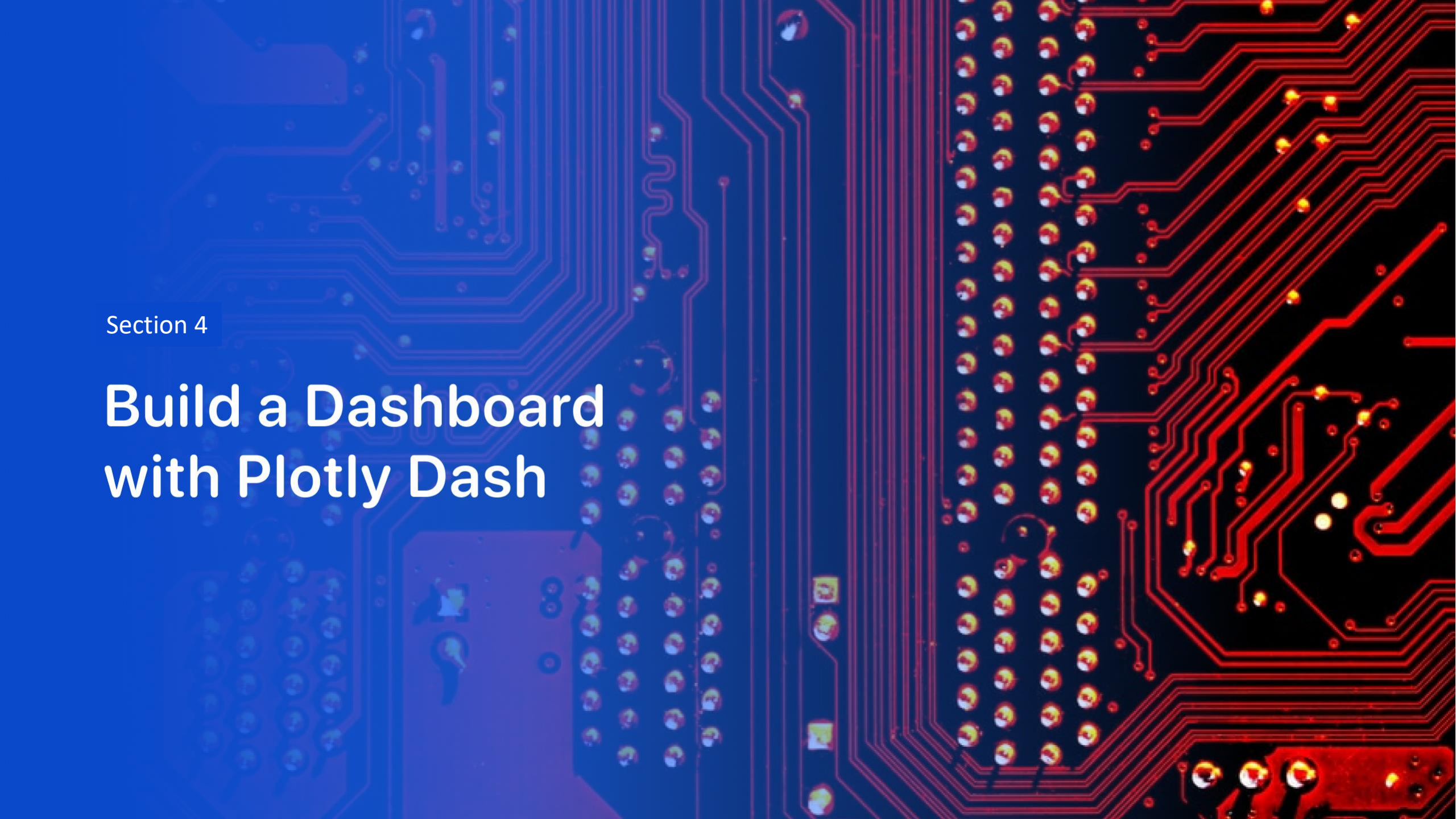




- Screenshots of the four launch sites zoomed in and displaying markers for each launch.
- Green markers show successful landings while red markers show failures.

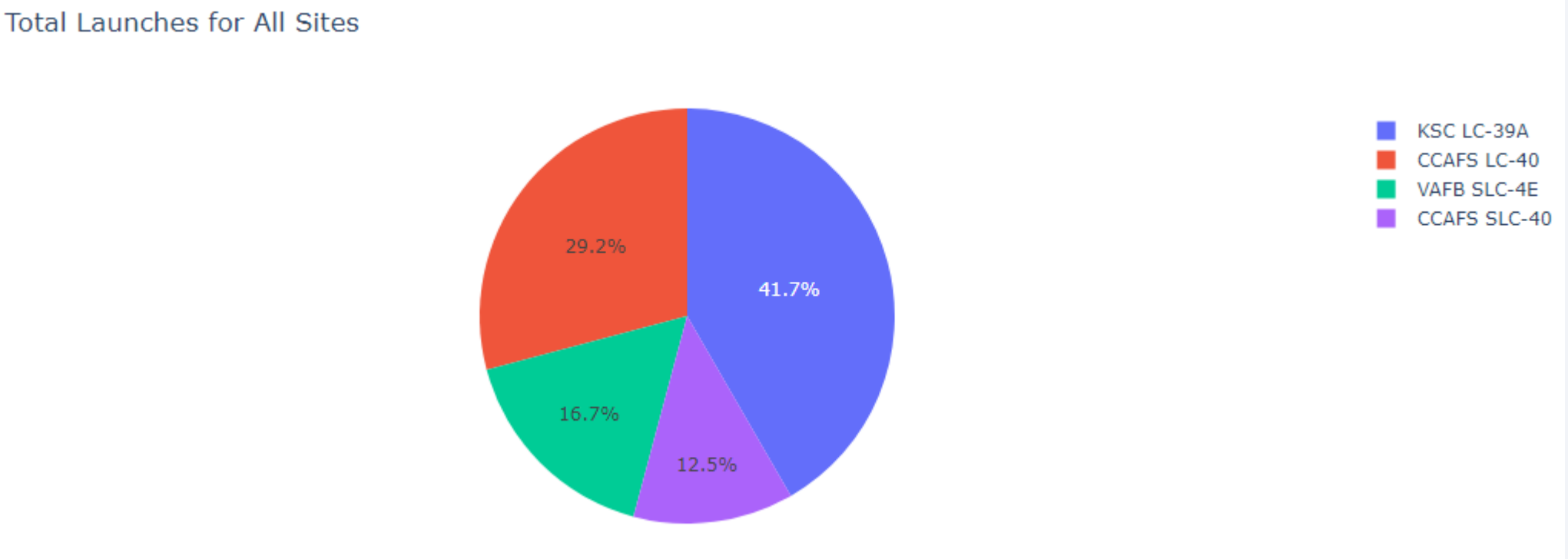
Launch Site Proximity to the Coast Line Visualized

- Folium was used to visualize lines from launch sites to different landmarks. In the screenshot below it shows the distance (0.9km) from launch site CCAFS SLC-40 to the coast line.


Section 4
Build a Dashboard with Plotly Dash

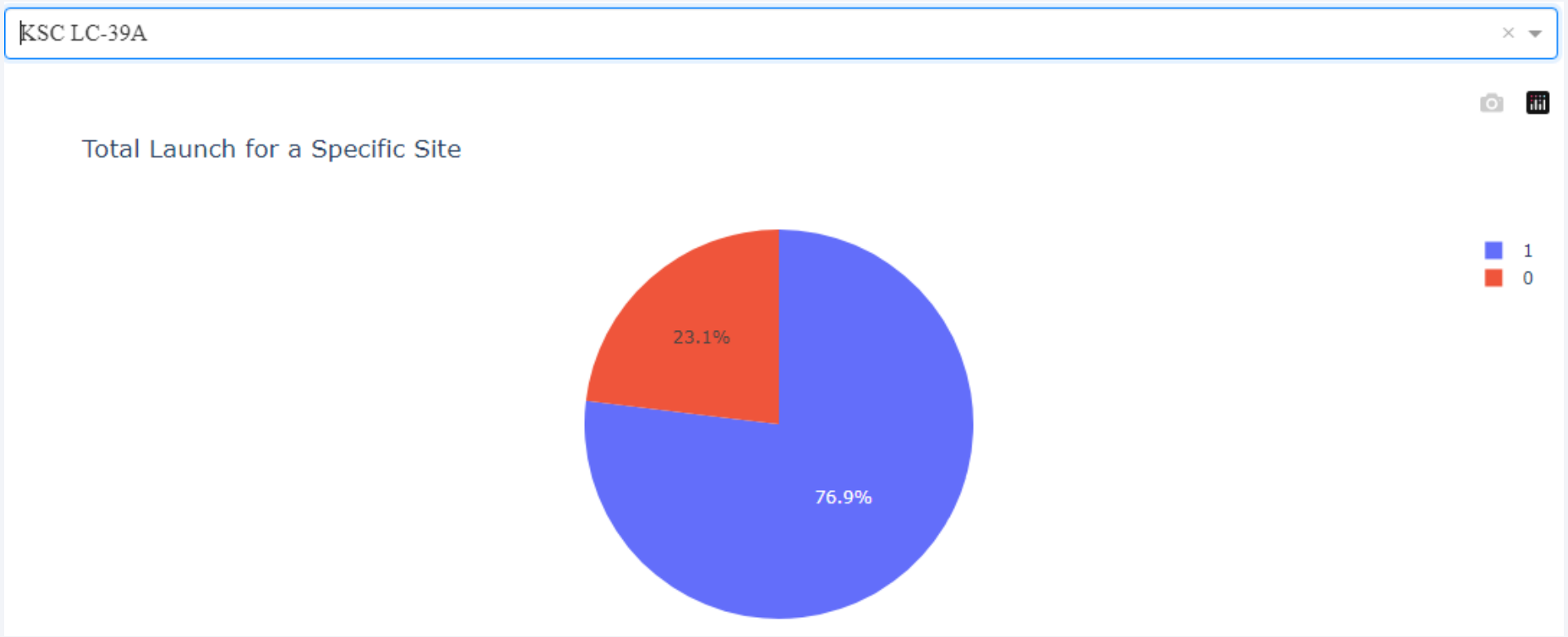
Success by Launch Site

- The interactive pie chart shows the total successful landings separated by each site and represented with a percentage. Based on this, site KSC LC-39A has the most successful landings.



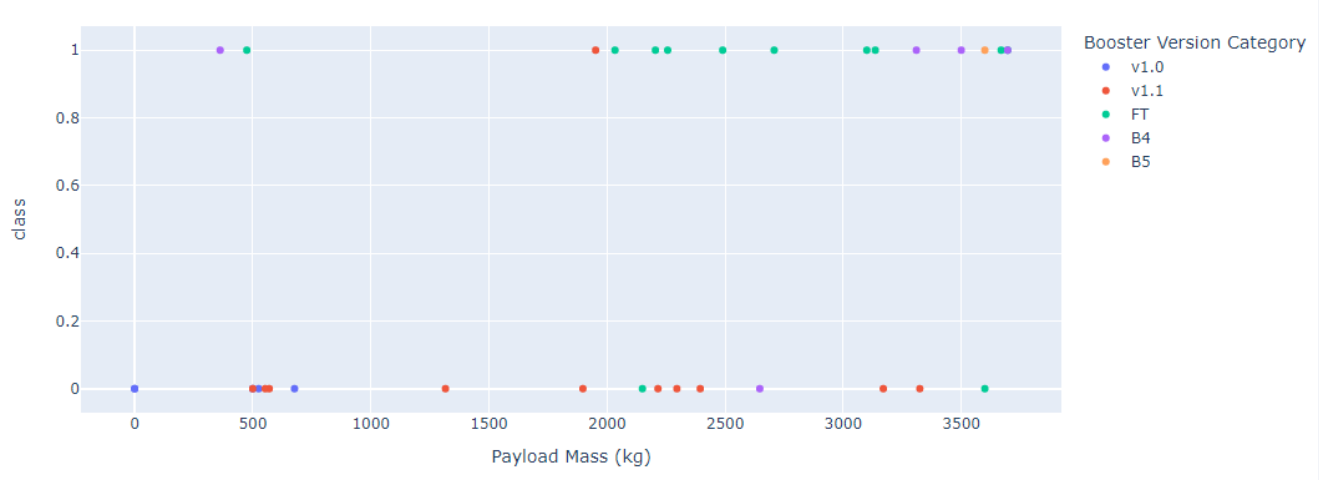
Launch Site with the Highest Success Rate

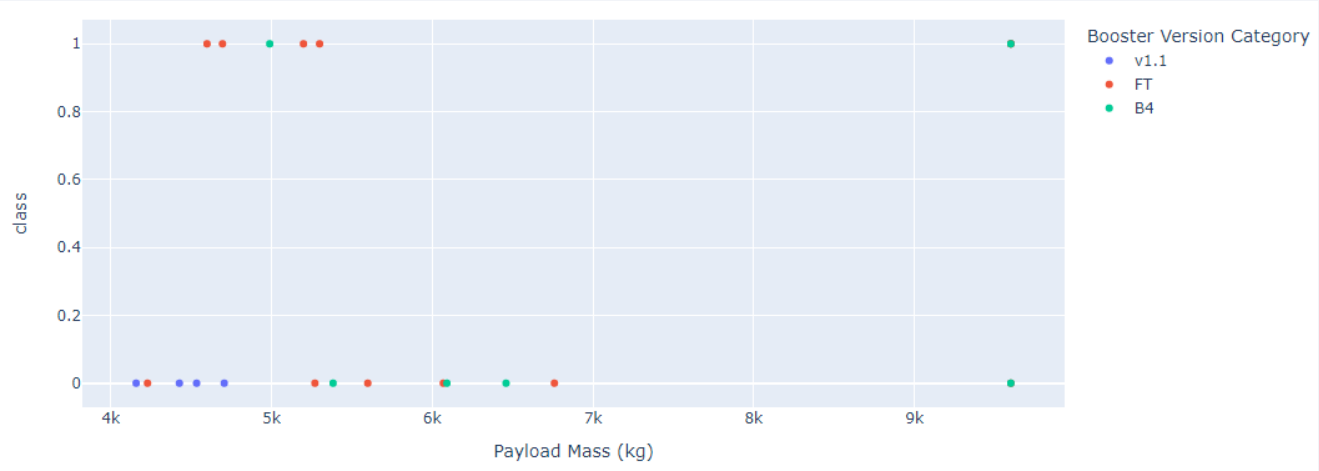
- KSC LC-39A has the highest success rate of all the launch sites with a success rate of 76.9%.



Success by Payload Mass

- The interactive Payload Mass slider allowed quick analysis of different payload sizes.
- From the screenshots it's fairly easy to see that lower payload mass sizes have a higher chance of success.





Section 5

Predictive Analysis (Classification)

Classification Accuracy

- From the code below it is determined that the best model is the Decision Tree.
- However, each of the models performed with very similar accuracy.

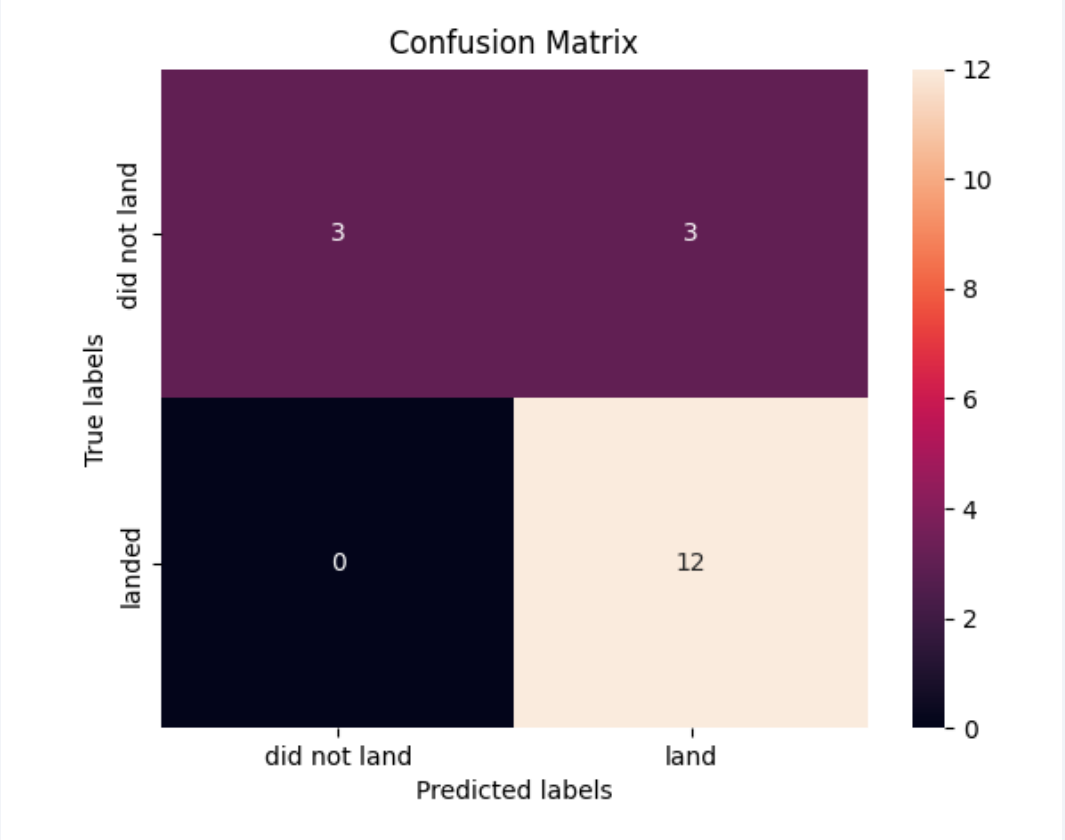
```
models = {'KNN':knn_cv.best_score_, 'DecisionTree':tree_cv.best_score_, 'LogisticRegression':logreg_cv.best_score_, 'SupportVectorMachine':svm_cv.best_score_}

bestalgorithm = max(models, key = models.get)
print('The best of the models is: ', bestalgorithm, ', with a score of ', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('The best parameters are: ', tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('The best parameters are: ', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('The best parameters are: ', logreg_cv.best_score_)
if bestalgorithm == 'SupportVectorMachine':
    print('The best parameters are: ', svm_cv.best_score_)
```

```
The best of the models is:  DecisionTree , with a score of  0.8875
The best parameters are:  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'best'}
```

Confusion Matrix

- The following is a confusion matrix for the best performing model: the decision tree.
- 3 True Negative predictions
- 3 False Positive predictions
- 0 False Negative predictions
- 12 True Positive predictions
- In total the model only made 3 inaccurate predictions of whether or not the landing would be a success.



Conclusions

- Launch site with the highest success rate was **KSC LC-39A**.
- Payload masses with the highest success rate were **4,000 kg and lower**.
- The orbit type with the highest success rate was **SSO**.
  - There were 3 other orbit types that had 100% success rate as well, but were skewed by only having 1 launch test each: ES-L1, HEO, and GEO.
- Landing success **increased over time** starting in **2013 until 2019** where it looks like it is possibly plateauing.
- The best model for predicting future successful landings is the **Decision Tree** model.

Appendix

- Week 1 – A – SpaceX Data Collection API
- Week 1 – B – Webscraping
- Week 1 – C – SpaceX Data Wrangling
- Week 2 – A – EDA with SQL
- Week 2 – B – EDA with Data Visualization
- Week 3 – A – Launch Site Location
- Week 3 – B – Plotly Dash
- Week 4 - A – SpaceX Machine Learning Prediction

Thank you!